INFORMATION AND CONTROL 29, 81–101 (1975)

On the Syntactic Structures of Unrestricted Grammars*
II. Automata

H. WILLIAM BUTTELMANN

Department of Computer and Information Science,
Ohio State University, Columbus, Ohio 43210

We define a generalization of the finite state acceptors for derivation structures and for phrase structures. Corresponding to the Chomsky hierarchy of grammars, there is a hierarchy of acceptors, and for both kinds of structures, the type 2 acceptors are tree automata. For $i = 0, 1, 2, 3$, the sets of structures recognized by the type i acceptors are just the sets of projections of the structures of the type i grammars, and the languages of the type i acceptors are just the type i languages. Finally, we prove that the set of syntactic structures of a recursively enumerable language is recursive.

INTRODUCTION

In Sections 1–5 (Part I, Buttelmann, 1975), we formalized the distinct notions of generative grammar and phrase structure grammar and the Chomsky hierarchy for each, and gave graph-theoretic definitions for the syntactic structures of both kinds of grammars, which we called, respectively, derivation structures and phrase structures. The material in Section 5 shows essentially that these two notions of syntactic structure are nonisomorphic in the sense that the natural correspondence between generative and phrase structure grammars does not, in general, preserve structural equivalence on their corresponding derivations. In this part we study automata on these structures.

With the exception of this introduction, Part II, including the numbering of sections, theorems, etc., and references to results in Part I, is written as a continuation of Part I.

* Portions of this paper appeared in the author's Ph.D. dissertation and in a paper by the author in the Proceedings of the Third Annual ACM Symposium on Theory of Computing.

6. Automata on Derivation Structures

In this and the next section we show that it is possible to define finite state automata on graphs in such a way that the automata are finite state acceptors for syntactic structures. In each case (derivation structures and phrase structures) the acceptors are generalizations of the now familiar tree automata (Doner, 1965, and Thatcher and Wright, 1965; see also Thatcher, 1970, and Rounds, 1970). For each case there is a hierarchy of acceptors corresponding to the hierarchy of grammars—i.e., corresponding to the type 0 generative grammars, there are type 0 derivation structure acceptors, for the type 1 generative grammars, type 1 derivation structure acceptors, for the type 1 phrase structure grammars, type 1 phrase structure acceptors, and so forth.

Following the terminology of tree automata, we call a set of structures accepted by a type i automaton a type i recognizable set. There are type 0 recognizable sets of derivation structures, etc. (The type 2 recognizable sets of derivation structures and the type 2 recognizable sets of phrase structures are both isomorphic to the recognizable sets of trees.) The chief results are that, for both derivation structures and phrase structures, the sets of structures defined by a type i grammar are related to the type i recognizable sets by projections. The proofs are constructive. In both cases, the type 2 acceptors are, in fact, tree automata, and readers familiar with tree automata will recognize these results as generalizations of the relationship between local sets (which are the type 2 derivation structures and type 2 phrase structures) and recognizable sets of trees (which are our type 2 recognizable sets).

In Sections 6 and 7 we restrict our attention to certain of the syntactic structures of a grammar: those which are the structures of so-called "complete" derivations.

Definition 6.1. Let $D = d\colon \sigma \Rightarrow \tau$ be a derivation of a grammar (gg or psg). D is a *complete* derivation iff $\sigma \in S$ and $\tau \in T^*$.

Definition 6.2. Let s be a syntactic structure (derivation structure/phrase structure) of a grammar (gg/psg) G. s is a *complete* syntactic structure (*complete* derivation structure/*complete* phrase structure) of G iff it is the structure of a complete derivation of G.

Clearly, if s is a complete syntactic structure of G, $l(\mathrm{dom}(s)) \in S$ and $l(\mathrm{fr}(s)) \in T^*$. For any *gg* G we denote by $S_c(G)$ the set of complete derivation structures of G, and for any psg G we denote by $P_c(G)$ the set of complete phrase structures of G.

DEFINITION 6.3. Let s be any derivation structure. s is *singly rooted* iff $|\text{dom}(s)| = 1$.

Every complete derivation structure is, of course, singly rooted. Now we return to the automata. Whereas classical finite state acceptors "read" a string of symbols, say one symbol at a time, derivation structure acceptors "read" a derivation structure by reading the strings of symbols which label the frontiers and dominators of the elementary derivation structures that make it up. The procedure is to configure the input mechanism so that it reads a string of labels on the dominator of some elementary derivation structure whose frontier has already been "recognized." Thus, the automaton accepts derivation structures by recognizing in 'bottom-up" sequence its elementary derivation structures.

DEFINITION 6.4. A *derivation structure automaton* (dsa) is a quadruple $A = (K, \Sigma, \delta, F)$, where

K is a finite set of *states*,

Σ is a finite set of *input symbols*,

$F \subseteq K$ is the set of *accepting states*,

δ is a function on $K^* \times \Sigma^+ \to \mathscr{F}(K^+)$, the finite subsets of K^+,[1] and where the following two restrictions hold:

(1) $\{\langle x, \sigma\rangle \mid \delta(x, \sigma) \neq \varnothing\}$ is finite,

(2) $\forall x \in K^*$ *and* $\forall \alpha \in \Sigma^+$, $\forall y \in \delta(x, \alpha)$, $\quad |y| = |\alpha|$.

In practice, we shall also follow the restriction: $\delta(\epsilon, \alpha)$ is defined $\Rightarrow |\alpha| = 1$. This last restriction results in no loss of generality, in terms of the sets of structures accepted, and it will simplify some of the definitions and proofs. Restrictions (1) and (2) are necessary to limit the computational "power" of the automata to derivation structures. In general, A is nondeterministic. If $\forall x \in K^*$ and $\forall \alpha \in \Sigma^+$, [either the cardinality of $\delta(x, \alpha)$ is 1 or $\delta(x, \alpha) = \varnothing$], then A is deterministic.

DEFINITION 6.5. Let $A = (K, \Sigma, \delta, F)$ be a dsa and $s = (N, E, \lessdot, \Sigma, l) = (\cdots (s(a) \circ^{n_1} \langle \alpha_1, \beta_1\rangle) \circ^{n_2} \cdots \circ^{n_n} \langle \alpha_n, \beta_n\rangle)$ be a singly rooted[2] derivation structure. A *run of A on s* is a map $r\colon N \to K$ such that:

[1] That is, for any set A, $\mathscr{F}(A)$ is defined to be the set $\{B \mid B \subseteq A$ and B is finite$\}$.

[2] If our grammars had axiom strings instead of axiom symbols, we would read arbitrary derivation structures instead of singly rooted ones, and we would accept with strings of accepting states instead of single accepting states.

(1) $\forall i \in \mathbf{n}$, $r(\mathrm{dom}(\langle \alpha_i, \beta_i \rangle)) \in \delta(r(\mathrm{fr}(\langle \alpha_i, \beta_i \rangle)), \alpha_i)$, and

(2) $\forall n \in \mathrm{fr}(s)$, $r(n) \in \delta(\epsilon, l(n))$.

In addition, r is an *accepting run* if it also satisfies the following condition:

(3) $r(\mathrm{dom}(s)) \in F$.

A derivation structure s is *accepted* by a dsa A just in case there is an accepting run of A on s. We shall denote the set of derivation structures accepted by A, $S(A)$. The language accepted by A, denoted $L(A)$, is the set of strings which are labels of the frontiers of the set $S(A)$, i.e., $L(A) = \{\sigma \mid \exists s \in S(A) \text{ such that } \sigma = l(\mathrm{fr}(s))\}$.

EXAMPLE 6.1. (cf. Example 3.4.). Consider the context sensitive language $\{0^n 1^n 2^n \mid n > 0\}$. We give a gg and a dsa for the language. Both have the same set of derivation structures. (Note that the dsa is deterministic.)

$$
\begin{array}{llll}
G: & S \to 0SA2 & & T = \{0, 1, 2\} \\
 & S \to 012 & & S = \{S\} \\
 & 2A \to A2 & & \\
 & 1A \to 11 & & \\
A: & \delta(\epsilon, 0) = \{k_0\} & & K = \{k_0, k_1, k_2, k_3, k_4\} \\
 & \delta(\epsilon, 1) = \{k_1\} & & \Sigma = \{0, 1, 2\} \\
 & \delta(\epsilon, 2) = \{k_2\} & & F = \{k_4\} \\
 & \delta(k_1 k_1, 1A) = \{k_1 k_3\} & & \\
 & \delta(k_3 k_2, 2A) = \{k_2 k_3\} & & \\
 & \delta(k_0 k_1 k_2, S) = \{k_4\} & & \\
 & \delta(k_0 k_4 k_3 k_2, S) = \{k_4\}. & &
\end{array}
$$

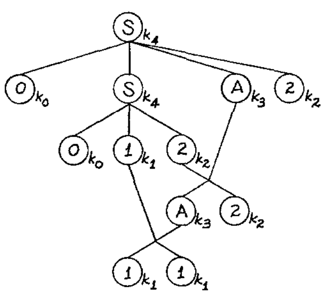

FIG. 15. Derivation structure with accepting run from Example 6.1.

The derivation structure in Fig. 15 is a derivation structure of the sentence 001122, and it is accepted by A. The state symbols noted beside each node illustrate an accepting run of A on the structure.

EXAMPLE 6.2. Let Σ be any finite alphabet containing a and b. The following dsa, A, accepts all Σ-labeled binary trees containing one or more subtrees of the form $\langle a, bb\rangle$. A is nondeterministic. In order to shorten the definition of A, we make use of the metavariables z, k, and q. z ranges over Σ; k and q range over K.

$$A: \quad \begin{aligned} \delta(\epsilon, z) &= \{z\} \\ \delta(kq, z) &\ni z \\ \delta(bb, a) &\ni q_A \\ \delta(q_A k, z) &\ni q_A \\ \delta(kq_A, z) &\ni q_A \\ \delta(q_A q_A, z) &\ni q_A . \end{aligned} \qquad \begin{aligned} K &= \Sigma \cup \{q_A\}, \quad \text{where } q_A \notin \Sigma \\ F &= \{q_A\} \end{aligned}$$

Figure 16 shows a member of $S(A)$, with an accepting run noted.

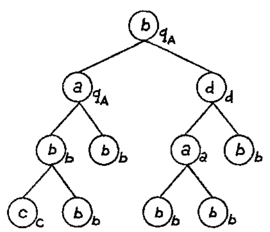

FIG. 16. A tree with accepting run by the dsa of Example 6.2.

EXAMPLE 6.3. Let $A = (K, \Sigma, \delta, F)$ be a dsa, and let $\delta(x, \alpha)$ be defined iff $|\alpha| = 1$ and $|x| \leqslant 1$. Then A is isomorphic to an ordinary (classical) finite state acceptor with state set $K \cup \{\epsilon\}$, start state ϵ, and accepting states F. Every $s \in S(A)$ has the form shown in Fig. 17, where $a_i \in \Sigma$. The set $T(A)$

FIG. 17. Form of derivation structures accepted by the dsa of Example 6.3.

of tapes accepted by the classical fsa is isomorphic to the set $S(A)$ of derivation structures accepted by A. $L(A)$ is just the subset of Σ containing those symbols which may be the final symbols in the tapes of the classical automaton.

Analogous to the Chomsky hierarchy (cf. Definition 1.3) there is a hierarchy of dsa:

DEFINITION 6.6. Every dsa is *type* 0. In addition, a dsa is:

(1) *type* 1 iff $\forall x \in K^+$ and $\forall \alpha \in \Sigma^+[\delta(x, \alpha) \neq \varnothing \Rightarrow |\alpha| \leqslant |x|]$,

(2) *type* 2 iff $\forall x \in K^+$ and $\forall \alpha \in \Sigma^+[\delta(x, \alpha) \neq \varnothing \Rightarrow |\alpha| = 1 \leqslant |x|]$,

(3) *type* 3 iff the state set K is partitioned into two nonnull disjoint blocks, K_1 and K_2, such that $K_1 = \{k \in K \mid \exists a \in \Sigma$ such that $k \in \delta(\epsilon, a)\}$, and $\forall x \in K^+$ and $\forall \alpha \in \Sigma^+[\delta(x, \alpha) \neq \varnothing \Rightarrow |\alpha| = 1 \leqslant |x| \leqslant 2$, and [either $x = \epsilon$ or $x \in K_1$ or $x \in (K_1 \times K_2)$], and $\delta(x, \alpha) \subseteq K_2]$.

It follows from the definition that every type i dsa is also a type $i - 1$ dsa, for $i \in 3$.

DEFINITION 6.7. Let $R \subseteq S_\Sigma{}^*$ be a set of derivation structures. R is a *type i recognizable set* (of derivation structures) iff there is a type i dsa A such that $R = S(A)$.

The relation between the derivation structures of gg's and dsa's is in terms of a projection:

DEFINITION 6.8. Let Γ and Σ be any two alphabets and let $\hat{\pi}: \Gamma \to \Sigma$ be a surjection. A *projection* on $S_\Gamma{}^*$ to $S_\Sigma{}^*$ is a function $\pi: S_\Gamma{}^* \to S_\Sigma{}^*$ defined by the following rule:

$$\forall s = (N, E, <\!\!\cdot, \Gamma, l) \in S_\Gamma{}^*, \qquad \pi(s) = (N, E, <\!\!\cdot, \Sigma, \hat{\pi} \circ l) \in S_\Sigma{}^*.$$

Thus, π is a map which relabels s according to a simple rule, but does not alter the structure of s.

The main results of this section are stated in Theorems 6.1, 6.2, and 6.5.

THEOREM 6.1. *For every generative grammar G there is a deterministic derivation structure automaton A such that $S_c(G) = S(A)$ and $L(G) = L(A)$. Furthermore, A is type i iff G is type i, for $i = 0, 1, 2, 3$.*

Proof. Given $G = (V, T, P, S)$, construct $A = (K, \Sigma, \delta, F)$ as follows:

$$K = \Sigma = V,$$
$$F = S;$$

δ is defined as follows: $\forall \beta \in V^+$ and $\forall \alpha \in V^+$

$$\delta(\beta, \alpha) = \begin{cases} \{\alpha\} & \text{if } \alpha \to \beta \in P, \\ \{\alpha\} & \text{if } \beta = \epsilon \text{ and } \alpha \in T, \\ \varnothing & \text{otherwise.} \end{cases}$$

A is indeed a deterministic dsa, and it has the same type as G. From Definition 3.3 any singly rooted $s \in S_\Sigma{}^*$ has the form

$$s = (\cdots(s(Z) \overset{\eta_1}{\circ} \langle \alpha_1, \beta_1 \rangle) \overset{\eta_2}{\circ} \cdots \overset{\eta_n}{\circ} \langle \alpha_n, \beta_n \rangle).$$

Construct the sequence

$$D = \langle \alpha_1 \to \beta_1, \eta_1 \rangle \cdots \langle \alpha_n \to \beta_n, \eta_n \rangle : Z \Rightarrow \sigma_n,$$

where for each $i \in \mathbf{n}$,

$$\sigma_i = \mathrm{fr}((\cdots(s(Z) \overset{\eta_1}{\circ} \langle \alpha_1, \beta_1 \rangle) \overset{\eta_2}{\circ} \cdots \overset{\eta_i}{\circ} \langle \alpha_i, \beta_i \rangle)).$$

Suppose $s \in S_c(G)$. We show that the map $r: N \to K$: $r(n) = l(n)$ is an accepting run of A on s, where N is the node set of s and l is the labeling function of s. Since $s \in S_c(G)$, by Proposition 3.2 $D \in D(G)$ and $s = s(D)$, and so for every $i \in \mathbf{n}$ $\alpha_i \to \beta_i \in P$, so $\delta(r(\mathrm{fr}(\langle \alpha_i, \beta_i \rangle)), \alpha_i) = \delta(\beta_i, \alpha_i) = \{\alpha_i\} = \{r(\mathrm{dom}(\langle \alpha_i, \beta_i \rangle))\}$. Furthermore, since s is complete, $Z = \alpha_1 \in S$ and $\mathrm{fr}(s) \in T^*$. Thus, for every $n \in \mathrm{fr}(s)$, $r(n) = l(n) \in T$ and so $\delta(\epsilon, r(n)) = \{r(n)\}$. Finally, $r(\mathrm{dom}(s)) = r(\mathrm{dom}(\langle \alpha_1, \beta_1 \rangle)) = \alpha_1 \in S = F$. Thus, $s \in S(A)$.

On the other hand, if $s \in S(A)$, then the accepting run of A on s is defined by $r(n) = l(n)$, for all $n \in N$, and from the construction of A, D must be a complete derivation of G. Then by Proposition 3.2, $s = s(D)$ and by Definition 6.2, $s \in S_c(G)$.

Since $S_c(G) = S(A)$, $L(G) = L(A)$. ∎

THEOREM 6.2. *Let A be any derivation structure automaton with alphabet Σ. There is a generative grammar G with alphabet V and there is a projection π: $S_V{}^* \to S_\Sigma{}^*$ such that $S(A) = \pi(S_c(G))$ and $L(A) = L(G)$. Furthermore, G is type i iff A is type i, for $i = 0, 1, 2, 3$.*

Proof. Given $A = (K, \Sigma, \delta, F)$ construct $G = (V, T, P, S)$ as follows:

$V = (K \times \Sigma) \cup \Sigma,$
$T = \{a \in \Sigma \mid \delta(\epsilon, a) \neq \varnothing\},$
$S = \{(q, a) \in (F \times \Sigma) \mid \exists x \in K^+ q \in \delta(x, a)\} \cup \{a \in \sigma \mid \delta(\epsilon, a) \cap F \neq \varnothing\};$

P is constructed as follows: $\forall x = q_1 q_2 \cdots q_n \in K^+$ and $\forall y = k_1 k_2 \cdots k_m \in K^+$ and $\forall \alpha = a_1 a_2 \cdots a_m \in \Sigma^+$ such that $y \in \delta(x, \alpha)$, construct the (finite) set $P_{x,\alpha,y} \subseteq (K \times \Sigma)^+ \times V^+$ as follows: $P_{x,\alpha,y} = \{A_1 A_2 \cdots A_m \to B_1 B_2 \cdots B_n \mid A_i = (k_i, a_i)$ for all $i \in \mathbf{m}$, and if A is not type 3:

$$B_i \in (\{q_i\} \times \Sigma) \cup \{b \in \Sigma \mid q_i \in \delta(\epsilon, b)\}, \qquad \forall i \in \mathbf{n},$$

and if A is type 3:

$$B_1 \in \{b \in \Sigma \mid q_1 \in \delta(\epsilon, b)\} \text{ and } B_2 \in (\{q_2\} \times \Sigma)\}.$$

Then $P = \bigcup_{x,\alpha,y} P_{x,\alpha,y}$.

Thus, for each definition: $y \in \delta(x, \alpha)$, the set $P_{x,\alpha,y}$ is formed as follows: the left-hand sides of all productions in $P_{x,\alpha,y}$ are the same:

$$(k_1, a_1)(k_2, a_2) \cdots (k_m, a_m);$$

the right-hand sides have the form $B_1 B_2 \cdots B_n$, where each B_i ranges over all pairs of the form (q_i, b), $b \in \Sigma$, and each B_i also ranges over all symbols $b \in \Sigma$ such that $q_i \in \delta(\epsilon, b)$. Typical productions of $P_{x,\alpha,y}$ are:

$$\begin{aligned} (k_1, a_1)(k_2, a_2) \cdots (k_m, a_m) &\to (q_1, a)(q_2, b) \cdots (q_n, c), \\ (k_1, a_1)(k_2, a_2) \cdots (k_m, a_m) &\to (q_1, a) \quad b \quad \cdots (q_n, c), \\ (k_1, a_1)(k_2, a_2) \cdots (k_m, a_m) &\to \quad a \qquad b \quad \cdots \quad c \quad , \end{aligned}$$

where $q_1 \in \delta(\epsilon, a), \ldots, q_n \in \delta(\epsilon, c)$.

We now prove that G has the required properties. Indeed, the sets V, T, P, and S are finite, and G is a generative grammar of the same type as A.

Define the surjection $\hat{\pi}: V \to \Sigma$ by $\hat{\pi}(q, a) = a$ and $\hat{\pi}(a) = a$. Then define the projection $\pi: S_V{}^* \to S_\Sigma{}^*$ according to Definition 6.8. In what follows we show that π is the projection of the theorem and that $S(A) = \pi(S_c(G))$.

Let $s \in S_c(G)$. If the weight of s is 0 then s is a single node n with $l(n) \in (S \cap T)$, and thus $\delta(\epsilon, l(n)) \cap F \neq \varnothing$, so $s \in S(A)$. If the weight of s is greater than zero, form the derivation structure $s' \in S(G)$ from s as follows: Let $s = (\cdots (s(Z) \circ^{\eta_1} \langle \alpha_1, \beta_1 \rangle) \circ^{\eta_2} \cdots \circ^{\eta_n} \langle \alpha_n, \beta_n \rangle)$. Define $s_0 = s(Z)$ and for each $i \in \mathbf{n}$ define $s_i = (s_{i-1} \circ^{\eta_i} \langle \alpha_i, \beta_i \rangle)$. Then define $s_0' = s_0$ and $s_i' = (s_{i-1}' \circ^{\eta_i'} \langle \alpha_i, \beta_i' \rangle)$, where $|\eta_i| = |\eta_i'|$ and if $\eta_i \alpha_i \xi_i = l(\mathrm{fr}(s_i))$ then

$\eta_i'\alpha_i\xi_i' = l(\mathrm{fr}(s_i'))$ for some ξ_i', and where $\alpha_i \to \beta_i'$ is that production in the same set of productions $P_{x,\alpha_i,y}$ as $\alpha_i \to \beta_i$ such that $\beta_i' \in (K \times \Sigma)^+$ and $\hat{\pi}(\beta_i') = \beta_i$. Observe that s and s' are identical except for their labels on the frontier. s' is constructed from s by relabeling as follows: if $n \in \mathrm{fr}(s)$, then $l(n) = a$ for some $a \in \Sigma$. In s', n will have the label (q, a) for some appropriate $q \in K$. So, s' is exactly like s except that all the leaves of s have labels in Σ and all the leaves of s' have labels in $(K \times \Sigma)$. Observe also that $\pi(s) = \pi(s')$; call it s''. Then

$$s'' = (\cdots(s(Z) \overset{\hat{\pi}(\eta_1)}{\circ} \pi(\langle\alpha_1, \beta_1\rangle)) \overset{\hat{\pi}(\eta_2)}{\circ} \cdots \overset{\hat{\pi}(\eta_n)}{\circ} \pi(\langle\alpha_n, \beta_n\rangle)).$$

Consider the map $r: N_s \to K$: $r(n) = q$, where $l_{s'}(n) = (q, a)$. For every $i \in \mathbf{n}$, let $\hat{\pi}(\alpha_i) \to \hat{\pi}(\beta_i) \in P_{x,\hat{\pi}(\alpha_i),y}$. Then $r(\mathrm{dom}(\pi(\langle\alpha_i, \beta_i\rangle))) = y \in \delta(x, \hat{\pi}(\alpha_i)) = \delta(r(\mathrm{fr}(\pi(\langle\alpha_i, \beta_i\rangle))), \hat{\pi}(\alpha_i))$. Also, $\forall n \in \mathrm{fr}(s'')$ there exists an eds$\langle\alpha, \beta\rangle$ in s'' such that $n \in \mathrm{fr}(\langle\alpha, \beta\rangle)$, and from the construction of the set $P_{x,\alpha,y}$ containing the production $\alpha \to \beta$, $r(n) \in \delta(\epsilon, l_s(n)) = \delta(\epsilon, l_{s''}(n))$. Finally since $s \in S_c(G)$, $r(\mathrm{dom}(s'')) \in F$, since from the definition of s, $l(\mathrm{dom}(s'')) = l(\mathrm{dom}(s)) \in (F \times \Sigma)$. Thus, r is an accepting run of A on $s'' = \pi(s)$.

On the other hand, let $s \in S(A)$. If the weight of s is zero, then s has a single node n and $\delta(\epsilon, l(n)) \cap F \neq \varnothing$, so $l(n) \in (S \cap T)$ and $s \in S_c(G)$. If the weight of s is greater than zero, let $r: N \to K$ be an accepting run of A on s. Let s have labeling function l. Form the derivation structure s' with labeling function l' by relabeling s as follows:

$$l'(n) = \begin{cases} (r(n), l(n)) & \text{if } n \in (N - \mathrm{fr}(s')), \\ l(n) & \text{if } n \in \mathrm{fr}(s'). \end{cases}$$

Observe that $s = \pi(s')$. We now show that $s' \in S_c(G)$ by exhibiting a complete derivation D of G such that $s' = s(D)$. From Definition 3.3 there is a sequence of eds's$\langle\alpha_1, \beta_1\rangle, \langle\alpha_2, \beta_2\rangle, \ldots, \langle\alpha_n, \beta_n\rangle$ and a sequence of strings $\eta_1, \eta_2, \ldots, \eta_n \in \Sigma^*$ and a symbol $Z \in \Sigma$ such that

$$s = (\cdots(s(Z) \overset{\eta_1}{\circ} \langle\alpha_1, \beta_1\rangle) \overset{\eta_2}{\circ} \cdots \overset{\eta_n}{\circ} \langle\alpha_n, \beta_n\rangle).$$

For each $i \in \mathbf{n}$, let $\langle\alpha_i', \beta_i'\rangle$ be the eds formed from $\langle\alpha_i, \beta_i\rangle$ by relabeling $\langle\alpha_i, \beta_i\rangle$ with l', and let η_i' be the string obtained from η_i as follows: if γ_i is the string of nodes in N^* such that $l(\gamma_i) = \eta_i$, then $\eta_i' = l'(\gamma_i')$. Finally let $Z' = l'(\mathrm{dom}(s))$. Then

$$s' = (\cdots(s(Z') \overset{\eta_1'}{\circ} \langle\alpha_1', \beta_1'\rangle) \overset{\eta_2'}{\circ} \cdots \overset{\eta_n'}{\circ} \langle\alpha_n', \beta_n'\rangle).$$

Now consider the sequence $D = \langle \alpha_1' \to \beta_1', \eta_1' \rangle \cdots \langle \alpha_n' \to \beta_n', \eta_n' \rangle: Z' \Rightarrow \sigma'$, where $\sigma' = l'(\mathrm{fr}(s'))$. For each $i \in \mathbf{n}$ there is a pair of strings $x_i, y_i \in K^+$ such that $y_i \in \delta(x_i, \alpha_i)$. Thus the set of productions P_{x_i, α_i, y_i} exists and from the construction of α_i' and β_i', $\alpha_i' \to \beta_i' \in P_{x_i, \alpha_i, y_i}$. Also, $\sigma' = l'(\mathrm{fr}(s')) = l(\mathrm{fr}(s))$, and so for each symbol a occurring in σ', $\delta(\epsilon, a)$ is defined, and from the construction of G, $\sigma' \in T^+$. Finally, $Z' = l'(\mathrm{dom}(s')) = \delta(q, Z)$ for some $q \in F$ and clearly there is an $x \in K^+$ such that $q \in \delta(x, Z)$, so $Z' \in S$. Thus D is a complete derivation of G, and by Proposition 3.2 and Definition 6.2, $s' = s(D) \in S_c(G)$.

Now consider any two derivation structures $s \in S_c(G)$ and $s' \in S(A)$, with labeling functions l and l', respectively, and such that $s' = \pi(s)$. From the construction of G, $l(\mathrm{fr}(s)) \in \Sigma^+$. But $l'(\mathrm{fr}(s')) = \pi(l(\mathrm{fr}(s)))$, so from the definition of π it must be the case that $l(\mathrm{fr}(s)) = l'(\mathrm{fr}(s'))$. Thus, $L(A) = L(G)$. ∎

In the proof of Theorem 6.2, we proved the following fact, which we now state as a lemma, since it will be useful in the proof of the next theorem.

LEMMA 6.3. *Let $A = (K, \Sigma, \delta, F)$ be a derivation structure automaton, let G be the generative grammar constructed by the construction in the proof of Theorem* 6.2, *and let π: $S_c(G) \to S(A)$ be the projection of Theorem* 6.2. *Let $s \in S_c(G)$ have nodes N and labeling function l.*

Then any map r: $N \to K$ defined by

$$r(n) = \begin{cases} q & \text{if } n \in (N - \mathrm{fr}(s)), \text{ where } l(n) = (q, a) \text{ for some } a \in \Sigma, \\ q & \text{if } n \in \mathrm{fr}(s), \text{ where } q \text{ is any member of } \delta(\epsilon, l(n)), \end{cases}$$

is well defined and is an accepting run of A on $\pi(s)$. ∎

The projection π of Theorem 6.2 is a surjection on $S_c(G) \to S(A)$. If however, the automaton is deterministic, then π is a bijection. We show this in the following

THEOREM 6.4. *Let A be a derivation structure automaton, let G be the generative grammar constructed as in Theorem* 6.2, *and let π: $S_c(G) \to S(A)$ be the projection of Theorem* 6.2. *If A is deterministic, then π is a bijection.*

Proof. Since $\pi(S_c(G)) = S(A)$, π is surjective. To show π is injective, assume it is not. Then $S_c(G)$ contains two derivation structures s' and s'' such that $s' \neq s''$ and $\pi(s') = \pi(s'')$. Let $s = \pi(s') = \pi(s'')$, let s, s', and s'' have labeling functions l, l', and l'', respectively, and let N be the set of nodes for s, s', and s''. From Lemma 6.3, the maps $r'\colon N \to K$ and $r''\colon N \to K$ defined by:

$$r'(n) = \begin{cases} q & n \in (N - \mathrm{fr}(s)), \text{ where } l'(n) = (q, a), \\ q & n \in \mathrm{fr}(s), \text{ where } q \in \delta(\epsilon, l'(n)), \end{cases}$$

$$r''(n) = \begin{cases} q & n \in (N - \mathrm{fr}(s)), \text{ where } l''(n) = (q, a), \\ q & n \in \mathrm{fr}(s), \text{ where } q \in \delta(\epsilon, l''(n)), \end{cases}$$

are accepting runs of A on s. But since s' and s'' are complete, $l(\mathrm{fr}(s')) \in \Sigma^*$ and $l''(\mathrm{fr}(s'')) \in \Sigma^*$, and so $l'(\mathrm{fr}(s')) = \pi(l'(\mathrm{fr}(s'))) = \pi(l''(\mathrm{fr}(s''))) = l''(\mathrm{fr}(s'')) = l(\mathrm{fr}(s))$. Thus, $r'(\mathrm{fr}(s)) = r''(\mathrm{fr}(s))$. Then, since A is deterministic, $r' = r''$. But in that case, for all $n \in N$, $l'(n) = l''(n)$, so $l' = l''$, and $s' = s''$, which contradicts the assumption. ∎

The following theorem is a stronger version of Theorem 6.1. We proved Theorem 6.1 separately because the automaton constructed there preserves the language of the grammar, whereas the automaton of this theorem cannot.

THEOREM 6.5. *Let G be a generative grammar with alphabet V, let Σ be any finite nonempty alphabet, and let $\pi\colon S_V{}^* \to S_\Sigma{}^*$ be any projection. There is a derivation structure automaton A with alphabet Σ such that $S(A) = \pi(S_c(G))$. Furthermore, A is type i if and only if G is type i, for $i = 0, 1, 2, 3$.*

Proof. Let $\hat{\pi}\colon V^* \to \Sigma^*$ be the usual homomorphic extension of $\hat{\pi}$, the surjection by which π is defined. Given $G = (V, T, P, S)$ construct $A = (K, \Sigma, \delta, F)$ as follows:

$$K = V;$$
$$\Sigma = \pi(V);$$
$$F = S;$$

δ is defined as follows: $\forall \beta \in K^*$ and $\forall \hat{\alpha} \in \Sigma^*$

$$\delta(\beta, \hat{\alpha}) = \begin{cases} \{\alpha \mid \hat{\alpha} = \hat{\pi}(\alpha) \text{ and } \alpha \to \beta \in P\} & \text{if } \beta \neq \epsilon, \\ \{\alpha \mid \hat{\alpha} = \hat{\pi}(\alpha) \text{ and } \alpha \in T\} & \text{if } \beta = \epsilon. \end{cases}$$

Observe that A is a nondeterministic dsa with the same type as G.

For $s \in S_c(G)$ let $s = (\cdots (s(Z) \circ^{\eta_1} \langle \alpha_1, \beta_1 \rangle) \circ^{\eta_2} \cdots \circ^{\eta_n} \langle \alpha_n, \beta_n \rangle)$. Then

$$\pi(s) = (\cdots (s(\hat{\pi}(Z)) \overset{\hat{\pi}(\eta_1)}{\circ} \langle \hat{\pi}(\alpha_1), \hat{\pi}(\beta_1) \rangle) \overset{\hat{\pi}(\eta_2)}{\circ} \cdots \overset{\hat{\pi}(\eta_n)}{\circ} \langle \hat{\pi}(\alpha_n), \hat{\pi}(\beta_n) \rangle).$$

We show that $\pi(s) \in S(A)$ by showing that the run $r\colon N \to K\colon r(n) = l(n)$, where N is the node set of s and $\pi(s)$ and l is the labeling function of s, is an accepting run of A on $\pi(s)$. Indeed, from the definitions of r and δ, for each i in $\mathbf{n}$, $r(\text{dom}(\langle\hat{\pi}(\alpha_i), \hat{\pi}(\beta_i)\rangle)) = \alpha_i \in \delta(\beta_i, \hat{\pi}(\alpha_i)) = \delta(r(\text{fr}(\langle\hat{\pi}(\alpha_i), \hat{\pi}(\beta_i)\rangle)), \hat{\pi}(\alpha_i))$. Also, since s is complete, $r(\text{dom}(\pi(s))) = Z \in S = F$, and for all $n \in \text{fr}(\pi(s))$, $l(n) \in T$, so $r(n) \in \delta(\epsilon, \hat{\pi}(l(n)))$. Thus, $\pi(s) \in S(A)$.

For $\hat{s} \in S(A)$ let $\hat{s} = (N, E, \lessdot, \Sigma, \hat{l})$, and let $r: N \to K$ be an accepting run of A on $\hat{s}$. Let $\hat{s} = (\cdots (s(\hat{Z}) \circ^{\hat{\eta}_1} \langle\hat{\alpha}_1, \hat{\beta}_1\rangle) \circ^{\hat{\eta}_2} \cdots \circ^{\hat{\eta}_n} \langle\hat{\alpha}_n, \hat{\beta}_n\rangle)$. Construct the sequence $s = (\cdots (s(Z) \circ^{\eta_1} \langle\alpha_1, \beta_1\rangle) \circ^{\eta_2} \cdots \circ^{\eta_n} \langle\alpha_n, \beta_n\rangle)$, where $Z = r(\text{dom}(\hat{s}))$, and $\forall i \in \mathbf{n}$, if $\hat{\eta}_i = \hat{l}(\gamma)$ then $\eta_i = r(\gamma)$ and $\alpha_i = r(\text{dom}(\langle\hat{\alpha}_i, \hat{\beta}_i\rangle))$ and $\beta_i = r(\text{fr}(\langle\hat{\alpha}_i, \hat{\beta}_i\rangle))$. Then $s = (N, E, \lessdot, V, r)$. We show that $s \in S_c(G)$ and $\hat{s} = \pi(s)$. From the definition of r, for each $i \in \mathbf{n}$,

$$\alpha_i = r(\text{dom}(\langle\hat{\alpha}_i, \hat{\beta}_i\rangle)) \in \delta(r(\text{fr}(\langle\hat{\alpha}_i, \hat{\beta}_i\rangle)), \alpha_i) = \delta(\hat{\beta}_i, \hat{\alpha}_i),$$

so from the definition of δ, $\alpha \to \beta \in P$ and $\hat{\alpha}_i = \hat{\pi}(\alpha_i)$. Thus, for every node $n \in (N - \text{fr}(s))$, $\hat{l}(n) = \hat{\pi}(r(n))$. Finally, since r is an accepting run, $r(\text{dom}(s)) = r(\text{dom}(\hat{s})) \in F = S$, and for all $n \in \text{fr}(s) = \text{fr}(\hat{s})$, $r(n) \in \delta(\epsilon, \hat{l}(n)) \subseteq T$ and $\hat{l}(n) = \hat{\pi}(r(n))$. Thus, $s \in S_c(G)$ and $\hat{l} = \hat{\pi} \circ r$, so $\hat{s} = \pi(s)$. ∎

COROLLARY 6.5.1. *Let $R \subseteq S_\Sigma^*$. R is a type i recognizable set of derivation structures if and only if there is a projection π and a type i generative grammar G such that $R = \pi(S_c(G))$, for $i = 0, 1, 2, 3$.*

Proof. Immediate from Theorems 6.2 and 6.5, and Definition 6.7. ∎

COROLLARY 6.5.2. *Let $L \subseteq \Sigma^*$. L is of type i language if and only if it is the language of a type i derivation structure automaton, for $i = 0, 1, 2, 3$.*

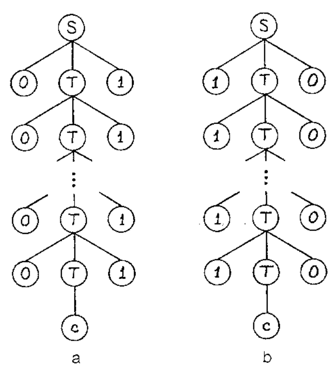

FIG. 18 (a)–(b). Forms of derivation structures for recognizable set R of Example 6.4.

Proof. Immediate from Theorems 6.1 and 6.2 and Definition 1.7. ▌

The following example illustrates the importance of the projection π.

EXAMPLE 6.4. Consider the set R containing all derivation structures having either the form pictured in Fig. 18(a) or the form pictured in Fig. 18(b). There is no cfg which defines R, but the type 2 derivation structure automaton, A, given below (a tree automaton) accepts R. Then by Theorem 6.2 there is a cfg and a projection such that R is the projection of the complete derivation structures of the cfg. G and π given below the automaton, are such a cfg and projection

$$A: \quad \begin{aligned} \delta(\epsilon, 0) &= \{k_0\} \\ \delta(\epsilon, 1) &= \{k_1\} \\ \delta(\epsilon, c) &= \{k_2\} \\ \delta(k_2, T) &= \{k_3, k_4\} \\ \delta(k_0k_3k_1, T) &= \{k_3\} \\ \delta(k_1k_4k_0, T) &= \{k_4\} \\ \delta(k_0k_3k_1, S) &= \{k_5\} \\ \delta(k_1k_4k_0, S) &= \{k_5\}. \end{aligned} \qquad \begin{aligned} K &= \{k_0, k_1, k_2, k_3, k_4, k_5\} \\ \Sigma &= \{0, 1, c, T, S\} \\ F &= \{k_5\} \end{aligned}$$

Figure 19 gives illustrations of typical accepting runs. The cfg G is:

$$\begin{aligned} S &\to 0T1 \\ S &\to 1U0 \\ T &\to 0T1 \\ U &\to 1U0 \\ T &\to c \\ U &\to c \end{aligned} \qquad \begin{aligned} T &= \{0, 1, c\} \\ S &= \{S\} \end{aligned}$$

and the projection is given by:

$$\begin{aligned} \hat{\pi}(S) &= S \\ \hat{\pi}(T) &= T \\ \hat{\pi}(U) &= T \\ \hat{\pi}(0) &= 0 \\ \hat{\pi}(1) &= 1 \\ \hat{\pi}(c) &= c. \end{aligned}$$

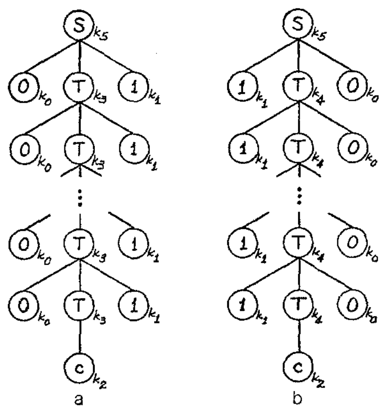

FIG. 19 (a)–(b). Typical accepting runs of dsa A of Example 6.4 on structures in recognizable set R.

EXAMPLE 6.5. Consider the csg's G_1 and G_2 given below.

$$R = S_c(G_1) \cup S_c(G_2)$$

is not defined by any csg, but there is a type 1 dsa which accepts R. Then, using Theorem 6.2, it is possible to construct a csg G_3 and define a projection π such that $R = \pi(S_c(G_3))$. We leave the definition of the type 1 dsa and the construction of G_3 to the reader.

$$\begin{array}{llll} G_1\colon & S \to 0SA2 & T = \{0, 1, 2\} & \\ & S \to 012 & S = \{S\} & \\ & 2A \to A2 & & \\ & 1A \to 11 & & \end{array} \qquad \begin{array}{llll} G_2\colon & S \to 2AS0 & T = \{0, 1, 2\} & \\ & S \to 210 & S = \{S\}. & \\ & A2 \to 2A & & \\ & \mathrm{A}1 \to 11 & & \end{array}$$

THEOREM 6.6. *Every recognizable set of derivation structures is recursive.*

Proof. Let K be the state set of a dsa accepting R, and let N be the node set of any $s \in S_\Sigma{}^*$. The set of maps of the form $r\colon N \to K$ is finite, and each can be checked in a finite length of time to see if it is an accepting run. ∎

If it seems surprising at first glance that the set of derivation structures of any r. e. is recursive, notice that the reason is that derivation structures contain more information than sentences—precisely that information necessary to tell, in general, whether a string is in a language.

7. Automata on Phrase Structures

The theory of phrase structure acceptors is analogous to the theory of derivation structure acceptors presented in the previous section. As in Section 4, we omit most of the explanation. Recall that the discussion is restricted to psg's of types 1, 2, and 3, to avoid dealing with the empty string in our graphs.

DEFINITION 7.1. A *phrase structure automaton* (psa) is a quadruple $A = (K, \Sigma, \delta, F)$, where

K is a finite set of *states*,

Σ is a finite set of *input symbols*,

$F \subseteq K$ is the set of *accepting states*,

δ is a partial function on $K^* \times \Sigma \times K^* \times K^* \to 2^K$, and where the following two restrictions hold:

(1) $\{\langle x, a, u, v\rangle \mid \delta(x, a, u, v) \neq \varnothing\}$ is finite,

(2) $\delta(\epsilon, a, u, v)$ is defined $\Rightarrow u = v = \epsilon$.

In general, A is nondeterministic. If either the cardinality of $\delta(x, a, y, v)$ is 1 or $\delta(x, a, u, v) = \varnothing$, then A is deterministic.

DEFINITION 7.2. Let $A = (K, \Sigma, \delta, F)$ be a psa and

$$p = ((N, E, \lessdot, \Sigma, l), k) = (\cdots(p(a_1) \underset{\mu_1-\nu_1}{\overset{\eta_1}{\circ}} \langle a_1, \beta_1\rangle) \underset{\mu_2-\nu_2}{\overset{\eta_2}{\circ}} \cdots \underset{\mu_n-\nu_n}{\overset{\eta_n}{\circ}} \langle a_n, \beta_n\rangle)$$

be a singly rooted phrase structure. A *run of A on p* is a map $r: N \to K$ such that:

(1) $\forall i \in \mathbf{n}$, $r(\mathrm{dom}(\langle a_i, \beta_i\rangle)) \in \delta(r(\mathrm{fr}(\langle a_i, \beta_i\rangle)), a_i, r(x_i), r(y_i))$, where $(x_i, y_i) = k(\mathrm{dom}(\langle a_i, \beta_i\rangle))$, and

(2) $\forall n \in \mathrm{fr}(p)$, $r(n) \in \delta(\epsilon, l(n), \epsilon, \epsilon)$.

In addition, r is an *accepting run* if it also satisfies the following condition:

(3) $r(\mathrm{dom}(p)) \in F$.

A phrase structure p is accepted by a psa A just in case there is an accepting run of A on p, and we denote by $P(A)$ the set of phrase structures accepted by A. The language accepted by A, $L(A)$, is the set of strings which are the labels of the frontiers of the set $P(A)$, i.e., $L(A) = \{\sigma \mid \exists p \in P(A)$ such that $\sigma = l(\text{fr}(p))\}$.

EXAMPLE 7.1. Again we give a grammar and an automaton for the language $\{0^n 1^n 2^n \mid n > 0\}$. This time they are phrase structural.

$$\begin{array}{lll} G: & S \to 0SA2 & T = \{0, 1, 2\} \\ & S \to 012 & S = \{S\} \\ & 2 \to X/\epsilon\_A & \\ & A \to Y/X\_\epsilon & \\ & X \to A/\epsilon\_Y & \\ & Y \to 2/A\_\epsilon & \\ & A \to 1/1\_\epsilon & \end{array}$$

$$\begin{array}{lrl} A: & \delta(\epsilon, 0) = \{k_0\} & K = \{k_0, k_1, k_2, k_3, k_4, k_5, k_6\} \\ & \delta(\epsilon, 1) = \{k_1\} & \Sigma = \{0, 1, 2\} \\ & \delta(\epsilon, 2) = \{k_2\} & F = \{k_6\} \\ & \delta(k_1, A, k_1, \epsilon) = \{k_3\} & \\ & \delta(k_2, Y, k_3, \epsilon) = \{k_5\} & \\ & \delta(k_3, X, \epsilon, k_5) = \{k_4\} & \\ & \delta(k_5, A, k_4, \epsilon) = \{k_3\} & \\ & \delta(k_4, 2, \epsilon, k_3) = \{k_2\} & \\ & \delta(k_0 k_1 k_2, S, \epsilon, \epsilon) = \{k_6\} & \\ & \delta(k_0 k_6 k_3 k_2, S, \epsilon, \epsilon) = \{k_6\}. & \end{array}$$

Figure 20 shows a phrase structure of the sentence 001122; it is accepted by A. The state symbols illustrate an accepting run.

DEFINITION 7.3. Every psa is *type* 1. In addition, a psa is:

(1) *type* 2 iff $\forall x \in K^+$ *and* $\forall a \in \Sigma$ *and* $\forall u, v \in K^*$ $[\delta(x, a, u, v)$ is defined $\Rightarrow u = v = \epsilon]$,

(2) *type* 3 iff the state set K is partitioned into two nonnull disjoint blocks, K_1 and K_2, such that $K_1 = \{k \in K \mid \exists a \in \Sigma$ such that $k \in \delta(x, a, \epsilon, \epsilon)\}$, and $\forall x \in K^+$ and $\forall a \in \Sigma$ and $\forall u, v \in K^*[\delta(x, a, v, u)$ is defined $\Rightarrow |x| \leqslant 2$, and [either $x = \epsilon$ or $x \in K_1$ or $x \in (K_1 \times K_2)]$, and $(x, a, \epsilon, \epsilon) \subseteq K_2]$.

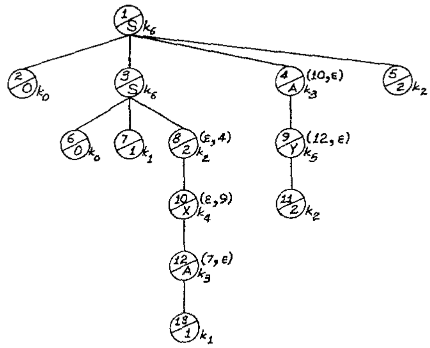

FIG. 20. Phrase structure with accepting run, from Example 7.1.

It follows from the definition that every type 3 psa is a type 2 psa and every type 2 psa is a type 1 psa.

DEFINITION 7.4. Let $R \subseteq P_\Sigma{}^*$ be a set of phrase structures. R is a *type i recognizable* set (of phrase structures) iff there is a type i psa A such that $R = P(A)$.

DEFINITION 7.5. Let Γ and Σ be any two alphabets and let $\hat{\pi}: \Gamma \to \Sigma$ be a surjection. A *projection* on $P_\Gamma{}^*$ to $P_\Sigma{}^*$ is a function $\pi: P_\Gamma{}^* \to P_\Sigma{}^*$ defined by the following rule:

$$\forall p = ((N, E, \lessdot, \Gamma, l), k) \in P_\Gamma{}^*, \quad \pi(p) = (N, E, \lessdot, \Sigma, \hat{\pi} \circ l), k) \in P_\Sigma{}^*.$$

π relabels p without changing its structure.

Theorems 7.1, 7.2, and 7.4 state the main results of this section.

THEOREM 7.1. *For every context sensitive phrase structure grammar G there is a deterministic phrase structure automaton A such that $P_c(G) = P(A)$ and $L(G) = L(A)$. Furthermore, A is type i iff G is type i, for $i = 1, 2, 3$.*

Proof. Given $G = (V, T, P, S)$ construct $A = (K, \Sigma, \delta, F)$ as follows:

$$K = \Sigma = V;$$
$$F = S;$$

δ is defined as follows: $\forall \beta, \mu, \nu \in V^*$ and $\forall a \in V$

$$\delta(\beta, a, \mu, \nu) = \begin{cases} \{a\} & \text{if } \ a \to \beta/\mu\_\nu \in P, \\ \{a\} & \text{if } \ \beta = \mu = \nu = \epsilon \text{ and } a \in T, \\ \varnothing & \text{otherwise.} \end{cases}$$

The proof that $P_c(G) = P(A)$ and $L(G) = L(A)$ is analogous to the proof of Theorem 6.1. We give the highlights. From Definition 4.3 every singly rooted $p \in P_V{}^*$ has the form

$$p = (\cdots(p(Z) \underset{\mu_1-\nu_1}{\overset{\eta_1}{\circ}} \langle A_1, \beta_1\rangle) \underset{\mu_2-\nu_2}{\overset{\eta_2}{\circ}} \cdots \underset{\mu_n-\nu_n}{\overset{\eta_n}{\circ}} \langle A_n, \beta_n\rangle).$$

From this form of p construct the sequence

$$D = \langle A_1 \to \beta_1/\mu_1-\nu_1, \eta_1\rangle \cdots \langle A_n \to \beta_n/\mu_n-\nu_n, \eta_n\rangle\colon Z \Rightarrow \sigma_n .$$

For $p \in P_c(G)$, from Proposition 4.1, $D \in D(G)$ and $p = p(D)$, and the map $r\colon N \to K\colon r(n) = l(n)$ is an accepting run of A on p. For $p \in P(A)$, the accepting run $r\colon N \to K\colon r(n) = l(n)$ shows that $D \in D(G)$, so by Proposition 4.1, $p = p(D)$ and by Definition 6.2, $p \in P_c(G)$. ∎

THEOREM 7.2. *Let A be any phrase structure automaton with alphabet Σ. There is a phrase structure grammar G with alphabet V and there is a projection $\pi\colon P_V{}^* \to P_\Sigma{}^*$ such that $P(A) = \pi(P_c(G))$ and $L(A) = L(G)$. Furthermore, G is type i iff A is type i, for $i = 1, 2, 3$.*

Proof. Given $A = (K, \Sigma, \delta, F)$ construct $G = (V, T, P, S)$ as follows:

$$\begin{aligned} V &= (K \times \Sigma) \cup \Sigma; \\ T &= \{a \in \Sigma \mid \delta(\epsilon, a, \epsilon, \epsilon) \text{ is defined}\}; \\ S &= \{(q, a) \in (F \times \Sigma) \mid \exists x \in K^+ q \in \delta(x, a, \epsilon, \epsilon)\} \\ &\quad\ \cup \{a \in \Sigma \mid \delta(\epsilon, a, \epsilon, \epsilon) \cap F \neq \varphi\}; \end{aligned}$$

P is constructed as follows: $\forall x = x_1x_2 \cdots x_m \in K^+$ and $\forall a \in \Sigma$ and $\forall q \in K$ and $\forall u = u_1u_2 \cdots u_n \in K^*$ and $\forall v = v_1v_2 \cdots v_r \in K^*$ such that $q \in \delta(x, a, u, v)$, construct the (finite) set $P_{x,a,u,v,q} \subseteq (K \times \Sigma) \times V^+ \times V^+ \times V^+$ as follows: $P_{x,a,u,v,q} = \{(q, a) \to B_1B_2 \cdots B_m/C_1C_2 \cdots C_n - D_1D_2 \cdots D_r\}$ if A is not type 3:

$$\begin{aligned} B_i &\in V \cap ((\{x_i\} \times \Sigma) \cup \{a \mid x_i \in \delta(\epsilon, a, \epsilon, \epsilon)\}), \\ C_i &\in V \cap ((\{u_i\} \times \Sigma) \cup \{a \mid u_i \in \delta(\epsilon, a, \epsilon, \epsilon)\}), \\ D_i &\in V \cap ((\{v_i\} \times \Sigma) \cup \{a \mid v_i \in \delta(\epsilon, a, \epsilon, \epsilon)\}), \end{aligned}$$

and if A is type 3:

$$\begin{aligned} B_1 &\in \{a \mid x_1 \in \delta(\epsilon, a, \epsilon, \epsilon)\} \\ B_2 &\in V \cap (\{x_1\} \times \Sigma)\}. \end{aligned}$$

Then $P = \bigcup_{x,a,u,v,q} P_{x,a,u,v,q}$.

The proof that G has the required properties is analogous to the proof of Theorem 6.2. The projection is the map defined by $\pi((q, a)) = a$ and $\pi(a) = a$. We leave the details to the reader. ▮

THEOREM 7.3. *Let A be a phrase structure automaton, let G be the phrase structure grammar constructed as in Theorem 7.2, and let $\pi: P_c(G) \to P(A)$ be the projection of Theorem 7.2. If A is deterministic, then π is a bijection.*

Proof. Analogous to Theorem 6.4. ▮

THEOREM 7.4. *Let G be a context sensitive phrase structure grammar with alphabet V, let Σ be any finite nonempty alphabet, and let $\pi: P_V{}^* \to P_\Sigma{}^*$ be any projection. There is a phrase structure automaton A with alphabet Σ such that $P(A) = \pi(P_c(G))$. Furthermore, A is type i if and only if G is type i, for $i = 1, 2, 3$.*

Proof. Given $G = (V, T, P, S)$ construct $A = (K, \Sigma, \delta, F)$ as follows:

$$K = V,$$
$$\Sigma = \pi(V),$$
$$F = S;$$

δ is defined as follows: $\forall \beta, \mu, \nu \in K^*$ and $\forall \hat{a} \in \Sigma$

$$\delta(\beta, \hat{a}, \mu, \nu) = \begin{cases} \{a \mid \hat{a} = \pi(a) \text{ and } a \to \beta/\mu\_\nu \in P\} & \text{if } \beta \neq \epsilon, \\ \{a \mid \hat{a} = \pi(a) \text{ and } a \in T\} & \text{if } \beta = \epsilon. \end{cases}$$

A is a nondeterministic psa with the same type as G.

For $p \in P_c(G)$ let

$$p = (\cdots(p(Z) \underset{\mu_1\_\nu_1}{\overset{\eta_1}{\circ}} \langle a_1, \beta_1 \rangle) \underset{\mu_2\_\nu_2}{\overset{\eta_2}{\circ}} \cdots \underset{\mu_n\_\nu_n}{\overset{\eta_n}{\circ}} \langle a_n, \beta_n \rangle).$$

Then

$$\pi(p) = (\cdots(p(\hat{\pi}(Z)) \underset{\hat{\pi}(\mu_1)\_\hat{\pi}(\nu_1)}{\overset{\hat{\pi}(\eta_1)}{\circ}} \langle \hat{\pi}(a_1), \hat{\pi}(\beta_1) \rangle) \underset{\hat{\pi}(\mu_2)\_\hat{\pi}(\nu_2)}{\overset{\hat{\pi}(\eta_2)}{\circ}}$$
$$\cdots \underset{\hat{\pi}(\mu_n)\_\hat{\pi}(\nu_n)}{\overset{\hat{\pi}(\eta_n)}{\circ}} \langle \hat{\pi}(a_n), \hat{\pi}(\beta_n) \rangle),$$

where $\hat{\pi}$ is the surjection of π, as in Definition 7.5. The proof that the map $r: N \to K: r(n) = l(n)$, where N is the node set of p and $\pi(p)$ and l is the labeling function of p, is an accepting run of A on p is analogous to the proof in Theorem 6.5.

For $\hat{p} \in P(A)$ let $\hat{p} = ((N, E, <\!\cdot, \Sigma, \hat{l}), k)$ and let $r: N \to K$ be an accepting run of A on $\hat{p}$. The proof that $p = ((N, E, <\!\cdot, V, r), k) \in P_c(G)$ and that $\hat{p} = \pi(p)$ is analogous to the proof in Theorem 6.5. ▮

COROLLARY 7.4.1. *Let $R \subseteq P_\Sigma{}^*$. R is a type i recognizable set of phrase structures if and only if there is a projection π and a type i phrase structure grammar G such that $R = \pi(P_c(G))$, for $i = 1, 2, 3$.*

Proof. Immediate from Theorems 7.2 and 7.4. ▮

COROLLARY 7.4.2. *Let $L \subseteq \Sigma^*$. L is a type i language if and only if it is the language of a type i phrase structure automaton, for $i = 1, 2, 3$.*

Proof. Immediate from Theorems 7.1 and 7.2 and Definition 1.7. ▮

THEOREM 7.5. *Every recognizable set of phrase structures is recursive.*

Proof. Analogous to Theorem 6.6. ▮

8. SPECULATIONS ON FURTHER RESEARCH

An obvious direction for further research on unrestricted syntactic structures is the extension of the notions of finite state transformations to these structures. Another direction is the investigation of the closure properties of the structures of context sensitive grammars and automata. Possibly the structures or the automata will lead to the proof of some new theorems about context sensitive languages, where relatively little is known and existing proofs are often tediously difficult. For one such set of results, see Stanat (1972).

In the introduction to this paper I claimed that a purpose of formal language theory is to provide a formal theory of syntax and a formal theory of semantics. I suggest that it is now possible to begin a theory of semantics for context sensitive and unrestricted grammars, based on the syntax theory in this paper. Indeed, some of the important concepts of such a semantic theory may already be in the literature (but not formalized into a comprehensive, integrated syntax–semantics theory)—for example, Knuth (1969 and 1971), and Benson (1971). An integrated syntax–semantics theory for context free grammars is presented formally in Buttelmann (1974).

Acknowledgment

The author is grateful to Jim Thatcher for spending much of his time to provide helpful comments on an earlier version of this paper.

Received: November 17, 1973; revised: November 27, 1974.

References

Benson, D. B. (1971), Syntax and semantics: a categorial view, *Inform. Contr.* **17**, 145–160.

Buttelmann, H. W. (1974), Semantic directed translation of context free languages, *Amer. J. Comput. Ling.* **2** (microfiche no. 7).

Buttelmann, H. W. (1975), On the syntactic structures of unrestricted grammars. I. Generative grammars and phrase structure grammars, *Inform. Cont.* **29**, 29–80.

Chomsky, N. (1956), Three models for the description of language, *IRE Trans. on Information Theory* **2**, 113–124. Reprinted *in* "Readings in Mathematical Psychology," (Luce, Bush, and Galenter, Eds.), Vol. 2, New York: Wiley, 1965.

Doner, J. E. (1965), Decidability of the weak second-order theory of two successors, Abstract 65T-468, *Not. Amer. Math. Soc.* **12**, 819.

Knuth, D. E. (1968), Semantics of context-free languages, *Math. Sys. Theory* **2**, 127–146.

Knuth, D. E. (1971), Examples of formal semantics, *in* "Proceedings of the Symposium on Semantics of Algorithmic Languages, Lecture Notes in Math. #188," (Engelor, Ed.), pp. 212–235, Springer-Verlag, New York.

Rabin, M. O. (1967), Mathematical theory of automata, "Mathematical Aspects of Computer Science," *Proc. Symp. Applied Math.* **19**, 173–175 (American Math. Soc., Providence, RI).

Rounds, W. C. (1970), Mappings and grammars on trees, *Math. Syst. Theory* **4**, 3.

Stanat, D. F. (1972), Approximation of weighted type 0 languages by formal power series, *Inform. Contr.* **21**, 344–381.

Tarski, A. (1936), Der Wahrheitsbegriff in den formalisierten Sprachen, *Studia Philosophica* **1**, 261–304.

Thatcher, J. W. and Wright, J. B. (1965), Generalized finite automata, Abstract 65T-469, *Not. Amer. Math. Soc.* **12**, 820.

Thatcher, J. W. (1967), Characterizing derivation trees on context-free grammars through a generalization of finite automata theory, *J. Comput Sys. Sci.* **1**, 317–322.

Printed in Belgium